PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : A63F 5/04, 9/24

A1

(11) International Publication Number: WO 99/65579

(43) International Publication Date: 23 December 1999 (23.12.99)

(21) International Application Number: PCT/AU99/00486

(22) International Filing Date: 17 June 1999 (17.06.99)

(30) Priority Data:
60/089,654 17 June 1998 (17.06.98) US

(71) Applicant *(for all designated States except US)*: ARISTOCRAT LEISURE INDUSTRIES PTY. LTD. [AU/AU]; 71 Longueville Road, Lane Cove, NSW 2066 (AU).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: BOND, Eugene, Thomas [US/US]; 6329 Lena King Avenue, Las Vegas, NV 89120 (US).

(74) Agent: F.B. RICE & CO.; 605 Darling Street, Balmain, NSW 2041 (AU).

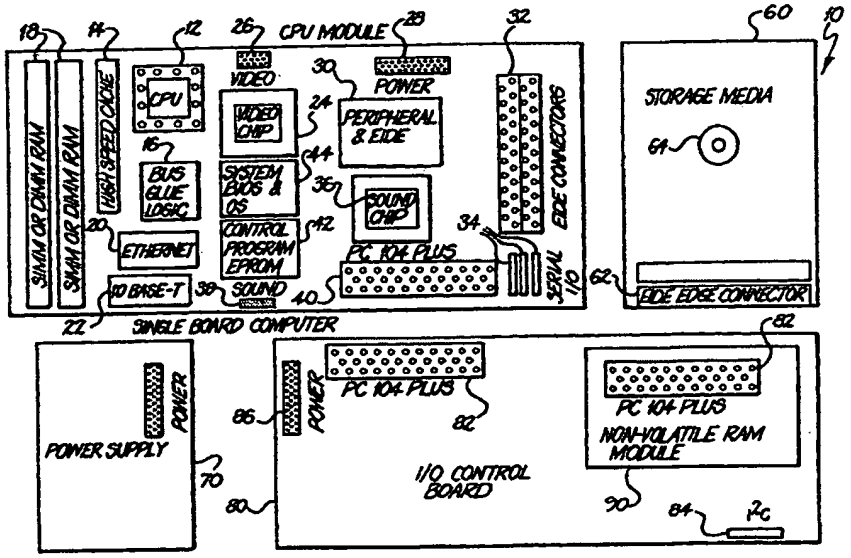
(81) Designated States: AU, JP, NZ, US, ZA, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: SOFTWARE VERIFICATION AND AUTHENTICATION

(57) Abstract

A device for controlling electronic gaming machines comprises a single board computer (SBC) having a microprocessor, memory means, storage means and a ROM (Read Only Memory). The ROM includes: a loader program; verification program; an authentication program; and a presentation program. Additional Mass storage media in communication with the SBC store pre–verified and approved gaming software (program files) and related data files, each of which have a verification signature appended to them. Prior to usage of the gaming software, the program file software or data file is retrieved by the loader program and checked by the verification program which compares the verification signature with a newly calculated verification signature. If the newly calculated signature matches the verification signature, the requested file is deemed to be intact (a validated image). The verification processes ensure that the file has been retrieved in its entirety and is free from corruption caused by storage media faults. If any corruption has occurred, the control device displays an error and the process is halted. After verification, all pending requests for authorization from authentication agents are processed by a queuing means. Each request includes a set of authentication instructions and a reply destination. After queuing, an authentication interpreter processes the validated image pursuant to the requester's instruction. The presentation program reports the resulting authentication identification to the requested destination which either acknowledges or refuses authentication. If acknowledged, the image is used or executed. If refused, an error is displayed and the process is halted.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Paso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

SOFTWARE VERIFICATION AND AUTHENTICATION

Background of the Invention

This invention relates to ensuring the compliance, integrity and authenticity of microprocessor-based gaming devices utilized in regulated gaming jurisdictions. These devices are commonly referred to as slot machines or video poker machines; however, more recent implementations have combined both aspects and offer a variety of games on a single machine referred to as a multi-game machine. A commonly configured multi-game machine would include a plurality of games such as: keno, poker, slots, blackjack and others. These games can be played separately or be combined together to form new games, games within games, thus pushing the limits of gaming software and hardware capabilities. As the complexity of these gaming devices increases, the difficulty of ensuring regulatory compliance increases.

Typical gaming machines of this type utilize a combination of mechanical devices, electronics, microprocessors and complex software to generate the gaming experience. Some of the common hardware components include a cabinet, handle, jackpot tower, coin acceptor, bill acceptor, credit meters, back-lit glass, reels, monitors, game doors, buttons, payout hoppers, lights and speakers. The electronics include many of the following components: microprocessor, (also referred to as a central processing unit ("CPU")), read only memory (ROM), random access memory (RAM), high speed data bus, peripheral logic chips for serial and parallel ports and driver circuitry for lamps, speakers, video and other devices. Typical software components include: power-up initialization, device drivers, game recovery, state machines (to monitor whether the game is in a state of active play, "sleeping" between active play or off), random number generator, payout routine, credit management, graphics engine, sound, game engine, game data, security, accounting and reporting functions.

In jurisdictions where gaming is legal, use of such gaming devices is regulated by law. This invention is designed to aid in complying with two kinds of gaming regulations: those requiring automated verification of the device's integrity, and those requiring a method of determining the authenticity of the device.

As will be described below, verification and authentication are separate processes. Verification of the gaming software is usually performed

before and during game play. Verification is done initially to make certain that the program code or other data stored in the gaming device is intact and functioning properly by methods known to those skilled in the art. In the case of verification, the gaming device's software is usually required to be check-summed or cyclic redundancy checked (CRC). During program execution (i.e., the course of game play) the software images are periodically re-checked in order to ensure that the storage media in which the program code/data is stored has not become corrupted. This periodic verification is important because media corruption has been known to generate erroneous jackpots.

Occasionally, the software is authenticated, which is typically a process carried out by a third party (other than the manufacturer or the operator/casino) representing the gaming enforcement agency that has jurisdiction over the device. Its purpose is to ensure that the software controlling the game has not been tampered with, and software authentication is usually required after a large jackpot has been obtained by a player. Authentication also verifies that the gaming software was previously examined and approved by the gaming agency in whose jurisdiction the jackpot occurred.

In addition, the casino likes to verify that the software running the game is legitimate particularly if the machine is not earning the expected amount of revenue or in response to player complaints about the behavior of a particular game.

In prior art devices, authentication is usually accomplished by one of two methods. Both methods require the opening of the game, the removal of CPU and the removal of software, typically stored in EPROMS, (Electronically Programmable Read Only Memory) from the CPU. Then, in the first method, the removed EPROMS are compared with a custodial (or master) set of EPROMS which have been kept in a secure location. If the comparison indicates that they are the same, the software is considered to be authentic. The second method involves plugging each EPROM into an electronic authentication device which generates an authentication identification (id) for the image resident in the EPROM. The resulting authentication ids are compared to previously recorded ids for those EPROMS. If they are identical, the software is labeled authentic.

Existing authentication methods are well-suited to prior art devices

which use ROM type storage; and which typically are stand-alone gaming machines. However, with the advent of new storage technologies, increased storage requirements of modern operating systems, and multi-game multimedia gambling devices involving a plurality of gaming machines in communication with each other, the prior art methods are no longer sufficient.

And though Silicon Gaming has invented a method for "authenticating" software stored in other media, it ignores the existing authentication paradigm presently accepted in gaming. Thus, there is a need for a means of verifying and authenticating software stored in modern media that is compatible with existing gaming regulations and practices. It is also believed that such methods should take into account the practice of relating software and modular functionality to EPROMS like prior art systems. The industry is comfortable with having a set of EPROMS for "System" software and a set for each model (comprising unique pay schedule, symbols, and/or play rules), or a set for each game in a multi-game environment. It is further thought that remote authentication is desirable to said agencies. Lastly, it is thought that a method of authentication that does not require the removal of gaming software from the machine is desirable to the operator.

Summary of the Invention

It is an object of the present invention to provide a device for use within a gaming machine, such as a slot machine or a multi-game machine, which allows for continuous verification of gaming software stored in modern media in a manner consistent to that which occurs in EPROM based prior art systems.

Still another object of the present invention is to provide a device for use within a gaming machine, such as a slot machine or a multi-game machine, which allows for verification of data files.

Still another object of the present invention is to provide a method that can be used within a gaming machine, that will allow software authentication without requiring the removal of gaming software (program files) from within the gaming machine.

Still another object of the present invention is to provide a method that allows for authentication of the gaming software (program files) without requiring removal of the central processing unit from the gaming machine.

Still another object of the present invention is to provide a method

that allows for authentication of the gaming software (program files) from a remote location.

Still another object of the present invention is to provide a method that allows for authentication of the gaming software (program files) according to specifications set forth by the I^2C.

Still another object of the present invention is to provide a method that allows for authentication of the gaming software (program files) from within the gaming machine.

Still another object of the present invention is to provide a method that can be used within a gaming machine, that will allow data file authentication without requiring the removal of the data files from the gaming machine.

Still another object of the present invention is to provide a method that allows for authentication of the data files without requiring removal of the central processing unit from the gaming machine.

Still another object of the present invention is to provide a method that allows for authentication of the data files from a remote location.

Still another object of the present invention is to provide a method that allows for authentication of data files according to specifications set forth by the I^2C.

Still another object of the present invention is to provide a method that allows for authentication of the data files from within the gaming machine.

Still another object of the present invention is to provide a method of relating gaming software stored in non-EPROM media as if it were embodied in EPROM media.

Still another object of the present invention is to provide a method of configuring gaming software stored in non-EPROM media as if the software were embodied in EPROM media.

Still another object of the present invention is to provide a method of reporting gaming software stored in non-EPROM media as if the software were embodied in EPROM media.

Still another object of the present invention is to provide a method of verifying gaming software stored in non-EPROM media as if it were embodied in EPROM media.

Still another object of the present invention is to provide a method of

authenticating gaming software stored in non-EPROM media as if it were embodied in EPROM media.

Still another object of the present invention is to provide a method of representing gaming software stored in non-EPROM media as if it were embodied in EPROM media.

Still another object of the present invention is to register the agent making the authentication request.

The present invention is a control system for electronic gaming machines which comprises a control means having a microprocessor, memory means, storage means, a means for operating the gaming machine, a plurality of program means and a plurality of data means, a ROM (Read Only Memory), a plurality of input/output means, a means for receiving power and a means for supplying power. The ROM includes means to verify and authenticate the program means and the data means, in response to a request from a requesting agent, which is external to the control system. The requesting agents can be located externally with respect to the gaming machine, or can be located within the gaming machine. The verification and authentication means include a loader program, a verification program, an authentication program, and a presentation program. Mass storage media in communication with the control means store pre-verified and approved gaming software (program files) and related data files, each of which has a unique verification signature (or identification means) appended to them. Prior to usage of the gaming software, a program file or data file is retrieved by the loader program and checked by the verification program, which compares the verification signature with a newly calculated verification signature. If the newly calculated signature matches the verification signature, the requested file is deemed to be intact (a validated image). The verification process ensures that the file has been retrieved in its entirety and is free from corruption caused by storage media faults. The control system also includes means to control the operation of the verification program and the gaming machine, depending on the outcome of the verification program and the authentication program, respectively. If any corruption has occurred, the control device displays an error message and the verification process is halted. After verification, all pending requests for authentication from authentication agents (requesting agents) are processed by a queuing means. Each request includes a set of authentication instructions and a reply

destination (a responder means). The responder means is external to the control system, and can be situated either externally with respect to the gaming machine, or internally in the gaming machine. After queuing, an authentication interpreter processes the validated image pursuant to the requester's instruction. The presentation program reports the resulting authentication identification to the requested destination which either acknowledges or refuses authorization. If acknowledged, the image is used; if refused, an error is displayed and the process is halted.

The present invention offers a number of benefits. First, the present invention allows one or more external authenticating agents to perform the authentication process. This results in better accountability to regulatory authorities because the manufacturer is not certifying its own authenticity. Second, each authenticating agent can use its own method (or multiple methods) of authentication using unique supplied authentication math instruction sets. Further, the same system can be used by the software manufacturer during the software release and/or upgrade process to ensure that the correct configuration of software has been installed on the gaming machine. This method is capable of distinguishing between media corruption and deliberate tampering of software components. In-house tampering or rigging of the software could also be detected if each department within the manufacturer has its own authentication instruction set; one group could easily check the work of another. Additional advantages include authentication which can be remotely accomplished; authentication requests are queued by the invention and can be flexibly scheduled; software can be managed as related groups through the V-PROM registry; and a wider range of gaming hardware can be used due to the hardware abstraction aspects of the registry.

The V-PROM (Virtual Programmable Read Only Memory) Registry aids in the retrieval and management of related stored binary information by logically grouping discrete data and program entities together as if they were stored in ROM type media.

The V-PROM Registry is a program/data directory which provides crucial information used in the management, retrieval and authentication of related programs and data sets. In prior art gaming inventions, related programs and data are typically stored in the same EPROM set. In such inventions, it is not uncommon to find a dozen system tasks or more stored

in the same EPROM set – a set is typically comprised of two EPROMS, the first containing the data stored in the odd- numbered addresses and the second containing the data stored in the even-numbered addresses. However, when using general purpose storage media such as a hard drive, the tendency, due to modern operating system design, is to store each process, task and data set as a separate discrete file. Using modular development techniques, it is not unusual to have thirty or more processes and several dozen data images stored on a hard drive in the form of files. Authentication of prior art systems is easily accomplished, being done at the EPROM level, resulting in one authentication id for each EPROM, a typical gaming device having one to twelve EPROMS. Whereas, the job of authenticating one hundred plus discrete files on a hard drive is a much more difficult and error-prone task. In order to maintain the same kind of one-to-many grouping characteristics naturally occurring in ROM style media, a relational database directory was designed to store logical EPROM-like grouping information, termed the V-PROM Registry. The Registry contains the location and media type of related stored programs and data sets, each group having a unique V-PROM name. As an added benefit, the V-PROM Registry abstracts the type of storage media from the authentication process thus allowing for the use of a wider variety of storage media.

The final step of the software installation phase is to update the directory of installed software in the V-PROM Registry, noting the images' storage locations, media types and defining logical groupings for related programs and associated data.

V-PROMS are logical containers that contain software and related data. V-PROMS can also contain other V-PROMS, thus, a given software configuration can be stored as a V-PROM containing a series of "smaller" V-PROMS which describe all of the software games to be installed for a specific operator/casino installation. Subsequently, by authenticating the "parent" V-PROM, the whole software installation can be verified.

Brief Description of the Drawings

The invention will be better understood by a Detailed Description of the Invention, with reference to the drawings, of which:

Figure 1 is a schematic diagram of the device of the present invention and related hardware components;

Figure 2 is a block diagram providing a conceptual overview of the in-

factory software installation process and the in-field software verification and authentication process; and

Figures 3A and 3B are block diagram describing the authentication process in greater detail than outlined in Figure 2.

Detailed Description of the Preferred Embodiment

Figure 1 illustrates the hardware components utilized in the preferred embodiment of this invention. Other hardware configurations are possible because of the modular nature of this design. The present invention is a control system 10 for a gaming machine. The gaming machine is not shown, but can be either a slot machine, a video poker machine, or a newer implementation which has combined aspects of both types of machines and offers a variety of games on a single machine referred to as a multi-game. A commonly configured multi-game machine includes a plurality of games such as: keno, poker, slots, blackjack and others. Typical gaming machines of this type utilize a combination of mechanical devices, electronics, microprocessors and complex software to generate the gaming experience. Some of the common hardware components used are as follows: a cabinet, handle, jackpot tower, coin acceptor, bill acceptor, credit meters, back-lit glass, reels, monitor, game door, buttons, payout hopper, lights and speakers.

The control system 10 (Fig. 1) comprises four major modules. The first module is a commercially available Single Board Computer (SBC). Second, Module 60 represents commercially available storage media. Third, module 70 is a commercially available power supply. The fourth module 80 is the custom manufactured Input Output (I/O) Control Board (the "IOCB"). A detailed description of the IOCB is found in NuGame's patent application Serial No.60,085892, for an Intelligent Input/Output control System, filed 18 May 1998, 1998. A fifth module 90 is a Non-volatile RAM storage board. All five components are in electronic communication with each other. SBC connects to storage medium 60 through either of an EIDE cable, or a SCSII cable in the case of a SCSII base SBC. SBC also is connected to the IOCB 80 and the non-volatile RAM storage board 90 through a stackable PC104 Plus connector. Power supply 70 supplies power to the other four components via power cables and connectors, components which are known to those skilled in the art. This regulated power supply provides power at the proper voltage and current that each component requires.

In detail, the SBC has the following components in electronic

communication with each other. The microprocessor (or CPU) 12 is a x86 CPU such as a Pentium™ or Pentium II™ processor with MMX™ or equivalent technology. CPU 12 is in communication with a high speed processor cache 14 for improved performance; glue logic chips 16 for driving high speed data busses and interfacing the processor 12 to other high speed components such as RAM, video processors, network chips, and I/O boards, High speed RAM sockets 18 typically DIMM or SIMM (Dual Inline Memory Module or Single Inline Memory Module, respectively) style; an integrated high speed network interface 20, such as, but not meant to be limited to, an Ethernet network connection; a network interface connector 22; an integrated high speed video processor, 24 video monitor connection; 26 are also components of the Single Board Computer 10. Other components of the SBC include power connector 28; a custom peripheral I/O cip 30 for driving a plurality of serial I/O, parallel I/O; and a disk subsets such as EIDE or SCSII. A plurality of EIDE or SCSII storage media connectors; 32, a plurality of serial and parallel I/O connections 34; an integrated sound chip 36; sound connection 38; and (9) stackable PC104 Plus connector 40 are also components of the SBC. It is important to note that this invention is not tied to a specific SBC or manufacturer. This is accomplished through the use of a general purpose operating system (OS) and a method of hardware abstraction. The manner of hardware abstraction is described in NuGame's patent application, Serial No.60/094068, for an I/O Interface and Device Abstraction, filed 24 July 1998, 1998. Thus, the present invention treats the SBC as a component which can be swapped or upgraded as new boards become commercially available.

The second module, also referred to as storage media module 60, represents the use of general use purpose mass storage media. The media represented in storage media module 60 can include: Hard disk drive units, CD-ROMs (Compact Disk-Read Only Memory), solid state storage devices, or other storage media used in the Personal Computer (PC) industry. This media is connected to the SBC through a PC industry standard interface such as EIDE, SCSII, or PCMCIA interface 62. Where spinning media is used, i.e., a hard disk drive unit, a CD-ROM drive unit, or a Digital Video Disk (DVD) drive unit, as examples, reference numeral 64 represents a drive spindle. Other elements that may be unique to a particular type of mass storage medium are not shown. Storage media 60 may also represent a file server

connected to the control system. Although the present invention can function without mass storage, it is specifically designed to exploit the advantages of such devices.

A custom I/O Control Board 80 is interfaced to the SBC through a stackable PC-104 Plus connector. This board is responsible for interfacing to all of the machine components utilized in a gaming device. This board also functions as a bridge to other intelligent I/O devices attached to the I^2C network at 84, an interface connection. Reference numeral 86 indicates a connector which connects the IOCB 80 with power supply 70. (A detailed description of the IOCB 80 is found in patent application Serial No.60/085892 for an Intelligent Input/Output Control System, filed 18 May, 1998.

Thus, the SBC and the IOCB 30 direct the operation of the various displays, controls, video, sound, and other peripherals of the gaming machine. The operation of SBC, IOCB 80 and the gaming machine(s) is directed by plurality of program files stored within the memory of the system's components.

The verification and authentication program included in the present invention has enhanced media verification, program/data management and authentication capabilities. These attributes enable the control system to better manage the large volume of software and data normally associated with a multi-game gaming device while ensuring the compliance, integrity and authenticity of each component. In addition, this invention also substantially aids in the reduction of software configuration errors that tend to accompany a device of this complexity.

Figure 2 is a diagram of a conceptual overview of the verification and authentication program 199. This verification and authentication program 199 is uniquely distinguished by interactions of three major components and the methods contained therein: a Loader 226 with media verification methods, a data/program directory 228 termed "V-PROM Registry" (Virtual Programmable Read Only Memory), and a distributed Authentication Engine. The Loader 226 functions to retrieve and verify the integrity of data and programs stored on a wide variety of media such as EPROMS, hard drivers, CD-ROMS, flash disks, fileservers, other ROMs such as PROM or EEPROM. The V-PROM Registry 228 aids in the retrieval and management of related stored binary information by logically grouping discrete data and program entities together. Contained within V-PROM Registry 228 are other program

files which include relational database and directory functions to perform this logical grouping of the program files and data files. The Authentication Engine is responsible for calculating and reporting authentication identifications (ids) according to instructions found in the pending authentication requests that are associated via V-PROM registry to the programs and related data being loaded. All three of these components interact in order to carry out requests for authentication as shown in Figure 2, and further described below.

The verification and authentication process to verify the integrity of the images being retrieved by the gaming software is a two-phase process and illustrated in Figure 2. The first phase, data/program preparation is accomplished by the In-Factory Software Installation Process 200. As its name suggests, the In-Factory Software Install Program is performed at the factory where the gaming machine is produced, prior to shipment. The second phase, software retrieval and verification process 220 is accomplished in the field by the Loader and Media Signature Verification routines located in the EPROM based Control Program stored on SBC. This phase verifies the integrity of the installed media, to make sure the program files and data files contained therein have not been damaged by faulty or defective storage media.

The first phase of the Software Installation configuration verification process 199 is an in-factory phase (also referred to as the data preparation phase 200) and is comprised of installing each processed program or data file 202 utilizing a Media Verification Signature Utility program 204. The media verification signature utility can be chosen from any one of a number of verification programs known to those skilled in the art. The output of this utility is a media verification signature 203. This signature 208 is calculated using either a cyclic redundancy check or a check sum using one of two common methods known to those skilled in the art. The complement 210 of the signature is appended to the end of its associated program or data set 206. The process is repeated for each program to be installed. Programs 208 are then installed on the recipient storage media 60. From storage media 60, the processed data and programs are transferred to the loader 226 for transfer to memory within the EPROM on SBC. The final step of the Installation phase is to update the directory of installed software in the V-PROM Registry 228, which notes the images' storage locations, media types and defines logical

groupings for related programs and associated data.

On power-up, the initialization software contained in the control program, not depicted, electronically checks for remotely attached authentication agents, if detected, they are interrogated for their authentication registry information, Name/ID of agency being represented, password if pre-registered and required, the registered agency is noted/logged in Non-volatile storage. An Agent can be installed on-the-fly and registered using NuGame's' Dynamic Hardware Linking technology. Note: Pre-registered agents are entered in the installation process. Agents may be electronic apparatus, or persons interacting with the machine through the console. Either may be asked to enter a password. Many agents can exist simultaneously. Agents can be remotely connected. Agents can be internally or externally located with respect to the machine.

Authentication responders are registered agents. Usually the Agent and the responder are the same person/Apparatus; however, a registered agent can request that an alternative responder agent be referenced, thus, allowing for a person (acting as an agent) to request an authentication which is presented to a handheld apparatus (acting as a second agent) such as laptop computer which acts as a responder. Many other possibilities exist, such as presenting authentication ids via a network connection so that two apparatus or persons are required, one local, one remote to accomplish the authentication process, thus enhancing the security of the process.

After all the gaming and related software has been installed onto the control system 10 by the in-factory install process 200, and the gaming machines are at their final destination, and being installed, or being periodically checked or operated, the second phase, the In-Field Software Verification and Authentication Process 220 is commenced. In the field (i.e. in the casinos, etc.) this process is performed during operation of the gaming device, from time-to-time, and as mandated by state regulatory agencies.

Phase two, the in-field, verification and authentication phase comprises two major steps – the first step being a verification step, and the second being an authentication step. The purpose of the verification step is to verify the integrity of the installed software (as program files and data files) to check that the various files, indicated by their images, have not been corrupted or altered or damaged because of faulty or defective media.

The verification step can be subdivided into the following steps, as

outlined in Figure 2. Whenever there is a request for a V-PROM, program or data set, such as may occur when a player activates the gaming machine to initiate game play the Loader routine 276, references the V-PROM Registry 228 and the loader routine 226 accesses the storage media and places the requested binary image of the data requested program file 208 in the CPU\12's main memory. The Media Signature Verification routine 230 then applies the same algorithm used in the data preparation phase (204) to the loaded image 208 which has an appended complimentary signature 211. Thus, using the check-sum method of verification, and due to the cancellation effect of previously appending the complement of the verification signature, the resulting check-sum value at 232 should be zero when processed again by the same algorithm. A non-zero result at 348 would indicate that the retrieved binary image was corrupted by a media fault resulting in the issuance of a command to halt the operation of the gaming device at 248. A result of zero at 232 indicates that the image has been wholly retrieved without corruption, and depending upon the priority commands given with the initial request, this information is placed in a queue and awaits authentication at 234. If the initial request included a priority command, the requested program is not queued, and can be executed or used immediately at 220.

If the verification method shows that the integrity of the media is intact, the authorization program is initiated. The second major component in the present invention is the V-PROM Registry 228. The V-PROM Registry 228 is a program data directory which provides crucial information used in the management, retrieval and authentication of related programs and data sets. In prior art gaming inventions, related programs and data are typically stored in the same EPROM set. An EPROM set is typically comprised of two EPROMS, the first containing the data stored in the odd numbered addresses and the second containing the data stored in the even numbered addresses.

In such prior art inventions, it is not uncommon to find a dozen system tasks or more stored in the same EPROM set.

Authentication of prior art systems is easily accomplished because it is done at the EPROM level, resulting in one authentication id for each EPROM, with a typical gaming device having one to twelve EPROMS. However, the newer gaming devices use general purpose storage media such as a hard drive, where the design of modern operating systems causes each process,

task and data set to be stored as a separate discrete file. Using modular development techniques, it is not unusual to have thirty or more processes and several dozen data images stored on a hard drive in the form of files as a result. Thus, the job of authenticating one hundred plus discrete files on a hard drive is a much more difficult and error prone task than authenticating tasks stored on EPROMS. In order to maintain the same kind of one-to-many grouping characteristics naturally occurring in ROM style media, a relational database directory was designed to store logical EPROM-like grouping information, termed the V-PROM Registry 228. Thus, the V-PROM is programmed to determine which program files and which data files are related, and to group them in a logical manner. This grouping program emulates the grouping methods that are characteristically found in ROM-type media, including ROMs, PROM, EPROM or EEPROM. The Registry contains the location and media type of related stored programs and data sets, each group having a unique V-PROM name. As an added benefit, the V-PROM Registry 228 abstracts the type of storage media from the authentication process thus allowing for the use of a wider variety of storage media. The third major component of the present invention is the Authentication Engine which is distributed into several smaller routines as shown in Figure 2. The Authentication Engine only acts upon authentication requests received from registered agents, a registered agent being an individual, a floor supervisor, system engineer or inspector authorized by a regulatory agency, the software manufacturer, the gaming machine manufacturer, authorized to perform the authentication process.

The software which controls the functioning of the present invention is stored in three different places. EPROM (42) contains the Control Program 220 illustrated in Figure 2. Storage device (44) contains the BIOS in control system 10. (Basic Input/Output System) and Operating System (OS) software (Figure 1). The type of storage device (44) varies from SBC to SBC, depending upon the manufacturer; in some cases the BIOS and Operating System software are stored in two separate devices in electronic communication. Software games (program files) and data (data files) are stored on Storage device 60. In some configurations, game software can be stored in a series of EPROMS attached to the PC-104 Plus bus 142 on the Non-volatile RAM module 90. In other configurations, game software can be stored on a file server attached to the network via connector 32. The file

server is not shown in the figures.

EPROM 42 contains the following software components (Figures 3A and 3B) software component 304C (the I^2C driver) is the driver to interface to the 10CB 80. Reference number 224 is an authentication requester queuing routine (see Figure 2 also). The data loader routine and a media verification routine is shown at reference number 226. The authentication interpreter is reference number 236, and it communicates with authentication presentation routine 240. Serial driver 304A and Network Driver 304B are part of the Operating System stored in the device 44 of the SBC.

Three of the authentication requestors 222, the serial connected authentication requestor 222A, the network connected authentication requestor 222B, and the I^2C connected authentication requestor 222C are located externally with respect to the control system 1, shown in Figure 1.

Similarly, several of the authentication responders, the serial connected authentication responder 242A, the network connected authentication responder 242B, and the I^2C connected authentication responder 242C, are externally located with respect to the control system 10 shown in Figure 1; the internal authentication request to 223 and the internal authentication responder 243 are optional, used by the manufacturer for internal authentication and stored in mass storage device 60.

An overview of the verification and authentication process 220 is presented in Figure 2 and described in the following few paragraphs. Figures 3A/3B describes the authentication process in greater detail. Identical reference numbers in Figure 2 and Figures 3A/3B correspond to an identical step in the process. For this reason, the reader should follow both Figures 2 and 3A/3B, as appropriate. Referring to Figures 3A/3B, requests are emitted by the external requesting agents 222 A-C which are in electronic communication with the present invention and transmitted to the authentication requestor queuing agent 224. Each of the connected authentication requestors, i.e., the serial connected authentication requestor 222A, the network connected authentication requestor 222B, and the I^2C connected authentication requestor 222C issue a request which is handled by serial driver 306A. As shown in Figures 3A and 3B, similar requests and drivers handle the other authentication requestors 222B and 222C. The optional internal authentication requestor 223 communicates directly with the authentication requestor queuing agent 224 by generating its own IPC

message. These requests for authentication originate from the "Authentication Requesters" 222A-C, or 223 . Properly formatted requests in the form of an (IPC Message 308) are queued by the Authentication Engine 224 in the Authentication Request Queue 234. Each request (IPC Message 308) contains a request code, scheduling information (i.e., whether it should be processed immediate, timed/periodic, or upon the occurrence of a triggering event/semaphore), an Authentication Responder Selection Code (which selects the authentication responder 242A-C or 243 responder which will make the determination of authenticity, indicating which program file or data file, each with an appropriate appended signature file, is to be retrieved and a set of math instructions (algorithm) for the Authentication Interpreter 236. The Authentication Interpreter 236 processes the requested V-PROM image by applying the math instructions contained in the queued request to each binary word of data comprising the retrieved V-PROM image 228A. The resulting value is termed the Authentication (ID) 238. The Authentication Presenter 240 then reports the generated Authentication ID 238 to one of the externally located Authentication Responder Agent 242A-242C, designated by the responder code of the queued request. The Authentication Responder Agent (242A-C, or 243) makes the determination as to whether the resulting calculated Authentication ID is consistent with that of an authentic, previously released, tested, inspected, and legally approved V-PROM image of the same name. The authentication identification for a given program file or data file is stored within the authentication responder agents, 242A-C or 243. If the generated authentication (id) 238 matches the authentication (id) stored in the authentication responder agent, 242A-C or 243, as appropriate, then the program file or data file is deemed to be authentic, and at 246, the operation of the gaming device is continued. If the generated authentication (id) 238 does not match that stored in the authentication responder agent, at 252 the operation of the gaming device is halted.

Every time a new V-PROM image is released by a manufacturer, each compliance agency inspects and test the V-PROM. After assuring jurisdictional compliance, the testing lab runs the authentication process on the approved V-PROM, using their own unique authentication request math instructions, resulting in a new Authentication ID 238 for that jurisdiction. This new id is recorded and distributed to Authentication Responder Agents for that jurisdiction. Authentication responses can be electronically

conveyed or manually entered by a field Agent.

Each individual authentication requestor (an agent) is initially registered with the system via a registration Inter-Process Communication, IPC, message 308.

The authentication queuing agent 224 cross examines the IPC request, checking the IPC message 308 for formatting errors. Properly formatted messages 348 are stored in the authentication request queue 234 and then acknowledged via a return IPC message. Errant requests (i.e., IPC messages that are improperly formatted or contain other errors detected by the authentication queuing agent 224) are also reported to the initiating requester (through a return IPC Message - not depicted). If the request is flagged for immediate processing 310 because it contains coding indicating it is be processed immediately, the loader 226 is informed via an IPC message 312. The loader 226 accesses the request stored in the queue 234, then, at 314 retrieves the requested V-PROM image 228A according to the registry 228 given, from storage media 228A. The authentication interpreter 236 is called via an IPC message 320 and processes the V-PROM image retrieved 228A using the request math instructions (algorithm) stored in the queue 234. The resulting authentication id 238 (Fig. 2) is sent from authentication interpreter 236 (Fig. 3B) via IPC message 322 to the authentication presenter 240. The presenter 240 routes the id to the responder agent 242, either the serial connected authentication responder 242A, the network connected authentication responder 242B, the I^2C connected authentication responder 242C, or the optional internal authentication responder 243 as designated by the contents of the queued request at 234. The responder 242A-C, or 243 sends either an approval or denial response back to the presenter 240. The IPC message 324 issued by the authentication presenter 240 is routed to the particular connected authentication responder (242A-C) by a corresponding driver (304A-C). Thus, as shown in Figs. 3A and 3B, serial driver 304A routes the IPC message and responds to serial connected authentication requestor 242A, the network driver 304B to 242B, etc. The internal authentication responder 243 receives its IPC message directly.

If the request is approved by the responder at 244, 240 deletes the queued request and at 246 continues normal execution; if not approved, at 252 the game is halted and an appropriate error message is displayed. The operation of the present invention can be illustrated by the following

example, involving an immediate authentication request for the "SYSTEM" V-PROM. A subroutine of the authentication engine known as the Authentication Request Agent 222 would queue at 224 the request, signaling the loader 226 to retrieve the named V-PROM data/program group 228, in this example, the "SYSTEM" V-PROM. The loader 226 retrieves and verifies the integrity of each component of the "SYSTEM" V-PROM at 230-234 handing off the verified data to the Authentication Interpreter 236. The Authentication Interpreter 236 computes an Authentication ID 238 for the V-PROM based on the instructions given in the request. The Authentication ID 238 is reported by the presenter 240 to an Authentication Responder 242A-C or 243 that was named in the queued request. The Authentication Responder 242A-C or 243 either replies to the Presenter 240 that the id is authentic at 244 or it replies that the id is not approved at 252. The Presenter 240, based on the response, either continues normal execution of the game at 246, or at 252 halts the device, displaying an authentication error. This error will persist until the system is reset through a manual process performed at the device.

It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

CLAIMS:

1. A control system for use with an electronic gaming device, the control system comprising:

a control means in electronic communication with the gaming device, the control means including:

a microprocessor means (a central processing unit ["CPU"]);

a memory means;

a storage means;

a means for operating the gaming device, the means for operating being stored in the memory means;

the memory means and the storage means further including a plurality of program means and a plurality of data means, and a method to verify the integrity of the program means and the data means (the verification method);

a plurality of input/output means;

a means for receiving power; and

a means for supplying power to the control system, the means for supplying power in electrical communication with the means for receiving power.

2. The control system as described in Claim 1, wherein each program means and each data means includes an identification means, such that each program means and each data means is uniquely identified (an established identification means).

3. The control system as described in Claim 2, wherein the control means further comprise a means for controlling one or more peripheral devices.

4. The control system as described in Claim 3, further compressing a second means for controlling one or more peripheral devices, the second means for controlling peripheral devices in communication with the control means.

5. The control system as described in Claim 3, wherein the first means for controlling peripheral devices is an Input/Output Control Board (IOCB).

6. The control system as described in Claim 5, further comprising a means for storing non-volatile memory.

7. The control system as described in Claim 6, wherein the storage means is chosen from the group consisting of a ROM, PROM, EPROM or EEPROM.

8. The control system as described in Claim 7, wherein the verification

method further includes a method for grouping the program means that are related, and for grouping the data means that are related the method for grouping emulating a method of grouping employed in storage media.

9. The control system as described in Claim 8, wherein the storage media whose grouping method is emulated is chosen from the group of storage media consisting of ROM, PROM, EPROM or EEPROM.

10. The control system as described in Claim 9, wherein the verification method further includes a method of abstracting the location of the program means, the data means and the storage means.

11. The control system as described in Claim 10, wherein the verification method, further includes means to compare the identification means of the requested program means or of the requested data means to the established identification means.

12. The control system as described in Claim 11, wherein the verification method further includes a method of controlling the operation of the gaming device in response to the verification of integrity of the program means or the data means.

13. The control system as described in Claim 12, wherein the controlling method includes a means of halting the verification method if the identification means of the requested program means or the requested data means does not match the established identification means of the program means or the data means.

14. The control system as described in Claim 13, wherein the verification method further includes a method to authenticate the retrieved program means or the retrieved data means.

15. The control system as described in Claim 14, wherein the control means effects the method to authenticate only after the integrity of the requested program means or the integrity of the requested data means has been verified.

16. The control system as described in Claim 1, wherein the method to verify the integrity of the program means and the data means further includes a method to authenticate the program means and the data means, the authentication method being activated in response to signals received from a requesting means.

17. The control system as described in Claim 15, wherein the requesting means is an authentication agent.

18. The control system as described in Claim 16, wherein the authentication agent is external to the control system and the gaming device, the authentication agent in communication with the control means.

19. The control system as described in Claim 16, wherein an authentication agent is external to the control system and is within the gaming device, the authentication agent in communication with the control means.

20. The control system as described in Claim 17 or Claim 18, wherein the authentication method further includes a method for registering the authentication agents.

21. The control system as described in Claim 18, wherein the signal received from the requesting means is an authentication request.

22. The control system as described in Claim 1, wherein the control means further includes a means for receiving the authentication requests.

23. The control system as described in Claim 1, wherein the authentication requests includes a signal to prioritize the authentication request.

24. The control system as described in Claim 23, wherein the control means further includes a method to queue the authentication requests, when more than one authentication request has been sent from the authentication agents.

25. The control system as described in Claim 24, wherein the control means further include a means of interpreting the authentication request.

26. The control system as described in Claim 25, wherein the means of interpreting the authentication request includes a means of generating an authentication identification (id) of the requested program means or data means.

27. The control system as described in Claim 26, wherein the control system further includes a responder means, the responder means being external to the control means and in electronic communication with the control means.

28. The control system as described in Claim 27, wherein the control means further includes a presenter means, the presenter means communicating the generated authentication id to the responder means.

29. The control system as described in Claim 28, wherein the control means and the responder means include a method of determining if the

generated authentication id is authentic, the responder means comparing the generated authentication id to the request, the generated authentication id deemed authentic if the generated authentication id matches the request.

30. The control system as described in Claim 29, wherein the generated authentication id is deemed not authentic if the generated authentication id does not match the request.

31. The control system as described in Claim 29 and 30, wherein the control means further includes a means of controlling the operation of the gaming device in response to the determination of authenticity of the requested program means or the requested data means.

32. The control system as described in Claim 31, wherein the controlling means includes means of halting the operation of the gaming device if the requested program means or the requested data means is deemed not authentic.

33. The control system as described in Claim 31, wherein the controlling means includes means of continuing the operation of the gaming device if the requested program means or the requested data means is deemed authentic.

34. The control system as described in Claim 9, wherein the storage means is a hard disk drive unit.

35. The control system as described in Claim 9, wherein the storage means is a CD-ROM unit.

36. The control system as described in Claim 9, wherein the storage means is a DVD unit.

37. The control system as described in Claim 9, wherein the storage means is a file server.

38. For use in an electronic gaming device, a method to verify the integrity of program means and the integrity of data means stored in a control system, the control system comprising:

a control means in electronic communication with the gaming device, the control system including;

a microprocessor means (a central processing unit ["CPU"]);

a memory means;

a storage means;

a means for operating the gaming device, the means for operating being stored in the memory means;

the memory means and the storage means further including a plurality of program means and a plurality of data means, each program means and each data means having an identification means, such that each program means and each data means is uniquely identified (an established identification means);

a plurality of input/output means;

a means for receiving power; and

a means for supplying power to the control system, the means for supplying power in electrical communication with the means for receiving power, the verification method comprising the steps of:

sending a request from a requesting means to the control system;

processing the request within the control system;

retrieving a requested program means or a requested data means from the storage means;

verifying the integrity of the requested program means or the requested data means by verification means which verify by comparing the identification means of the requested program means or the requested data means with the request, the integrity verified if the identification means matches the established identification means request; and

controlling the operation of the gaming device in response to the verification of integrity of the requested program means or the requested data means.

39. The method as described in Claim 39, further comprising the steps of halting the verification method of the identification means of the requested program means or the requested data means does not match the established identification means of the program means or the data means.

40. The method as described in Claim 39, further comprising a method to authenticate the retrieved program means or the retrieved data means.

41. The method as described in Claim 40, wherein the method to authenticate is effected only after the integrity of the requested program means or the integrity of the requested data means has been verified.

42. The method as described in Claim 41, wherein the requesting means is an authentication agent.

43. The method as described in Claim 42, wherein the method further includes a method for registering the authentication agent.

44. The method as described in Claim 43, wherein the request includes a

verification request and an authentication request.

45. The method as described in Claim 44, wherein the request further includes an authentication queuing request.

46. The method as described in Claim 45, wherein the request further includes registration means for the authentication agent.

47. The method as described in Claim 43, further including a method of abstracting the location of the program means, the data means, and the storage means.

48. The method as described in Claim 47, further including the step determining which of the program means are related, and determining which of the data means are related.

49. The method as described in Claim 48, further including the step of grouping the related program means, and grouping the related data means.

50. The method as described in Claim 49, wherein the grouping step emulates a method of grouping employed in storage media chosen from the group consisting of ROM, PROM, EPROM or EEPROM.

51. The authentication method as described in Claim 50, wherein the control means further includes a means for queuing the authentication requests.

52. The authentication method as described in Claim 51, further comprising the step of queuing the authentication requests, when more than one authentication request has been sent from the authentication agents.

53. The authentication method as described in Claim 52, wherein the control means further includes a means of interpreting the authentication request.

54. The authentication method as described in Claim 53, further comprising the step of interpreting the authentication request.

55. The authentication method as described in Claim 54, wherein the interpretation step includes the step of generating an authentication identification (id).

56. The authentication method as described in Claim 55, wherein the control means further includes a presenter means, the presenter means communicating the generated authentication id to a responder means.

57. The authenticating method as described in Claim 56, further comprising the step of determining if the generated authentication id is authentic, the responder means and the control means comparing the

generated authentication id to the request, the generated authentication id deemed authentic if the generated authentication id matches the request.

58. The authentication method as described in Claim 57, wherein the generated authentication id is deemed not authentic if the generated authentication id does not match the request.

59. The authentication method as described in Claim 57 or 58, further including the step of controlling the operation of the gaming device in response to the determination of authenticity of the requested program means or the requested data means.

60. The method as described in Claim 59, wherein the controlling step includes halting the operation of the gaming device if the requested program means or the requested data means is determined to be not authentic.

61. The method as described in Claim 59, wherein the controlling step includes continuing the operation of the gaming device if the requested program means or the requested data means is determined to be authentic.

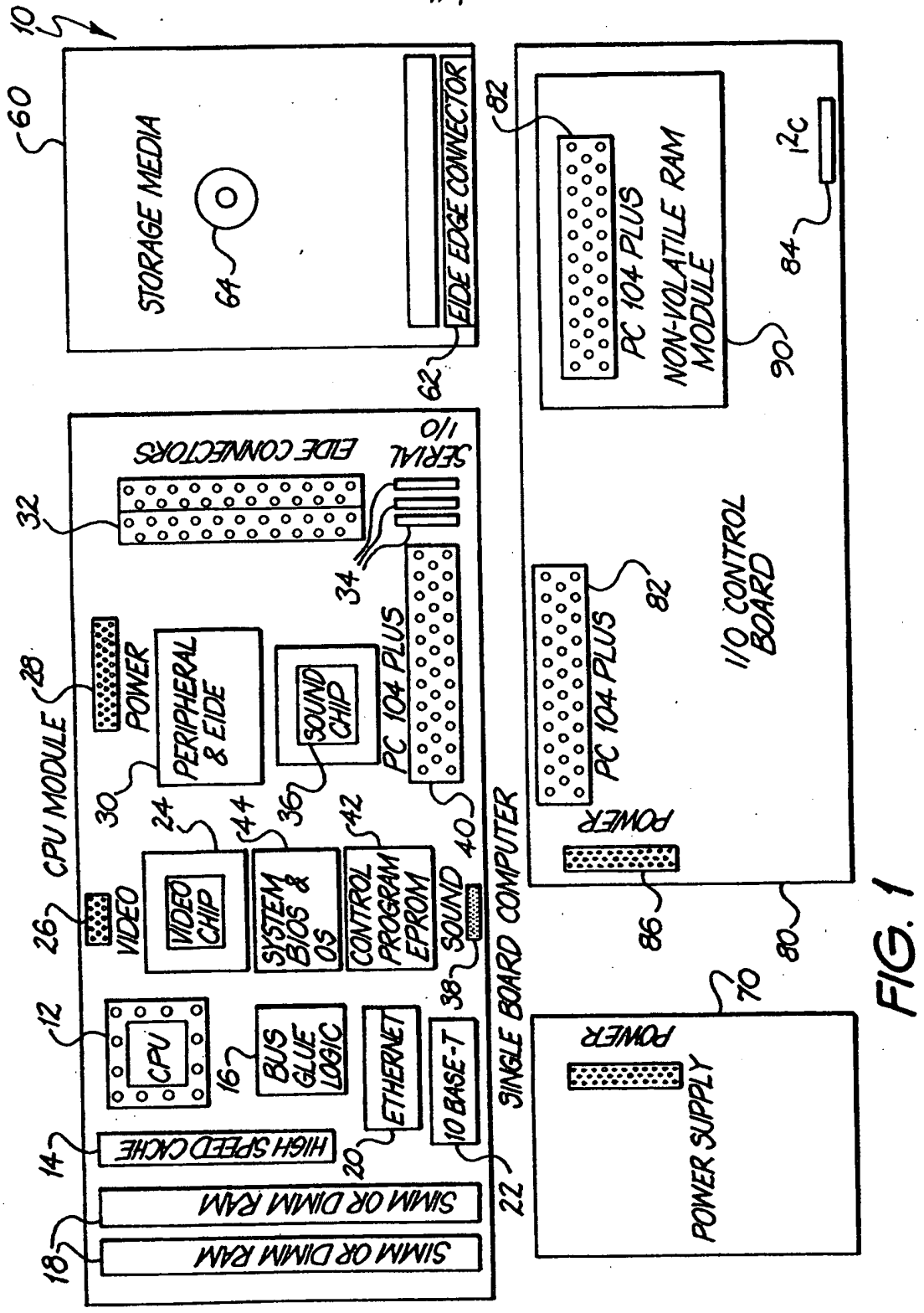

10
60
STORAGE MEDIA
64
EIDE EDGE CONNECTOR
62
82
PC 104 PLUS
NON-VOLATILE RAM MODULE
90
I2C
84
I/O CONTROL BOARD
SERIAL I/O
EIDE CONNECTORS
32
34
CPU MODULE
28
POWER
PERIPHERAL & EIDE
30
24
SOUND CHIP
44
36
PC 104 PLUS
42
40
82
PC 104 PLUS
26
VIDEO
VIDEO CHIP
SYSTEM BIOS & OS
CONTROL PROGRAM EPROM
SOUND
38
POWER
86
80
12
CPU
16
BUS GLUE LOGIC
ETHERNET
10 BASE-T
SINGLE BOARD COMPUTER
70
POWER
POWER SUPPLY
14
HIGH SPEED CACHE
20
22
SIMM OR DIMM RAM
SIMM OR DIMM RAM
18

FIG. 1

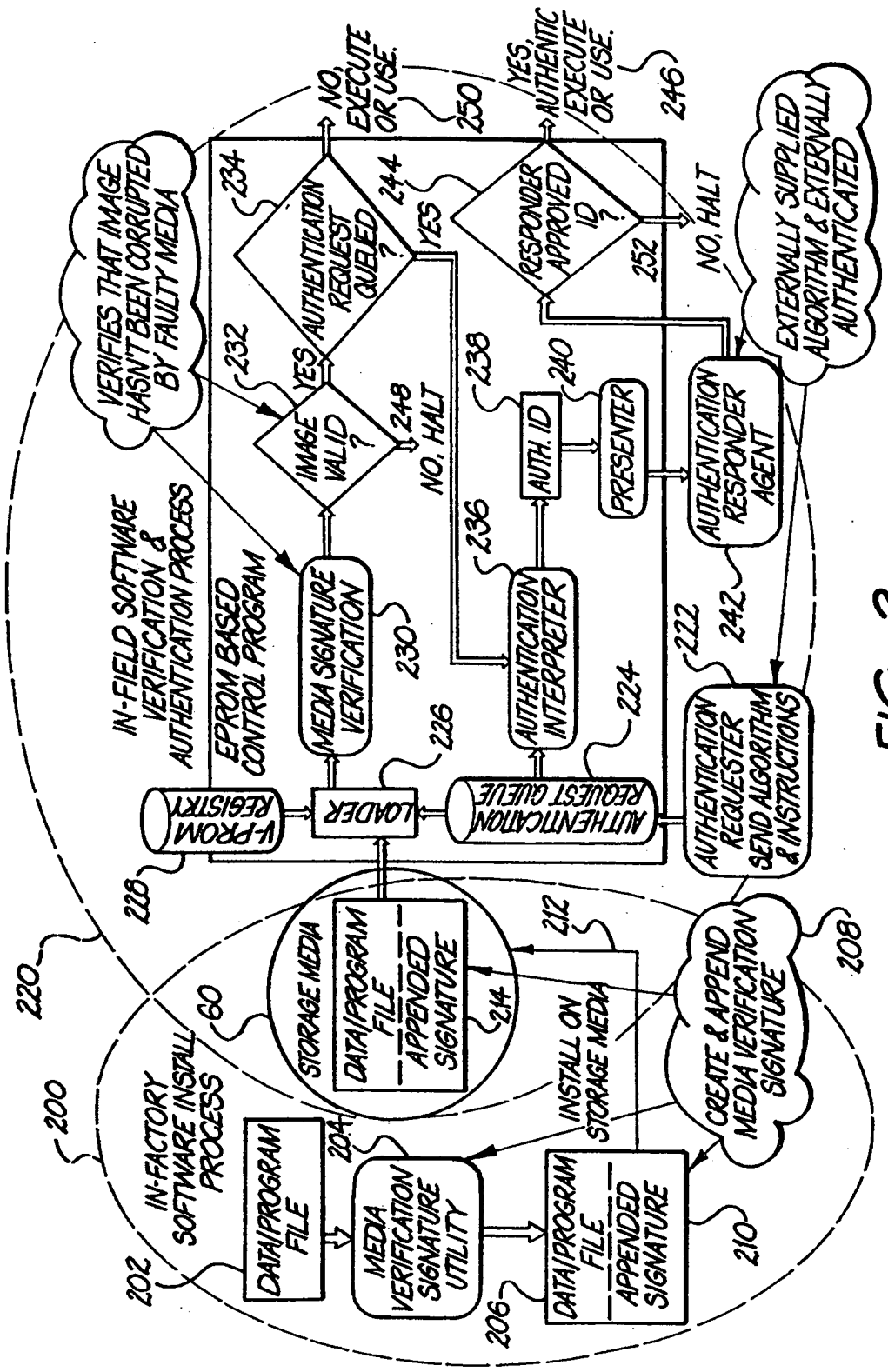

200
IN-FACTORY SOFTWARE INSTALL PROCESS
202
DATA/PROGRAM FILE
204
MEDIA VERIFICATION SIGNATURE UTILITY
206
DATA/PROGRAM FILE APPENDED SIGNATURE
210
INSTALL ON STORAGE MEDIA
212
STORAGE MEDIA
60
DATA/PROGRAM FILE APPENDED SIGNATURE
214
220
CREATE & APPEND MEDIA VERIFICATION SIGNATURE
108
IN-FIELD SOFTWARE VERIFICATION & AUTHENTICATION PROCESS
EPROM BASED CONTROL PROGRAM
V-PROM REGISTRY
228
LOADER
226
MEDIA SIGNATURE VERIFICATION
230
VERIFIES THAT IMAGE HASN'T BEEN CORRUPTED BY FAULTY MEDIA
IMAGE VALID ?
232
NO, HALT
248
AUTHENTICATION REQUEST QUEUED ?
234
YES
AUTHENTICATION REQUEST QUEUE
224
AUTHENTICATION INTERPRETER
236
AUTH. ID
238
PRESENTER
240
RESPONDER APPROVED ID ?
244
NO, HALT
252
NO, EXECUTE OR USE.
250
YES, AUTHENTIC EXECUTE OR USE.
246
AUTHENTICATION RESPONDER AGENT
242
EXTERNALLY SUPPLIED ALGORITHM & EXTERNALLY AUTHENTICATED
AUTHENTICATION REQUESTER SEND ALGORITHM & INSTRUCTIONS
222

FIG. 2

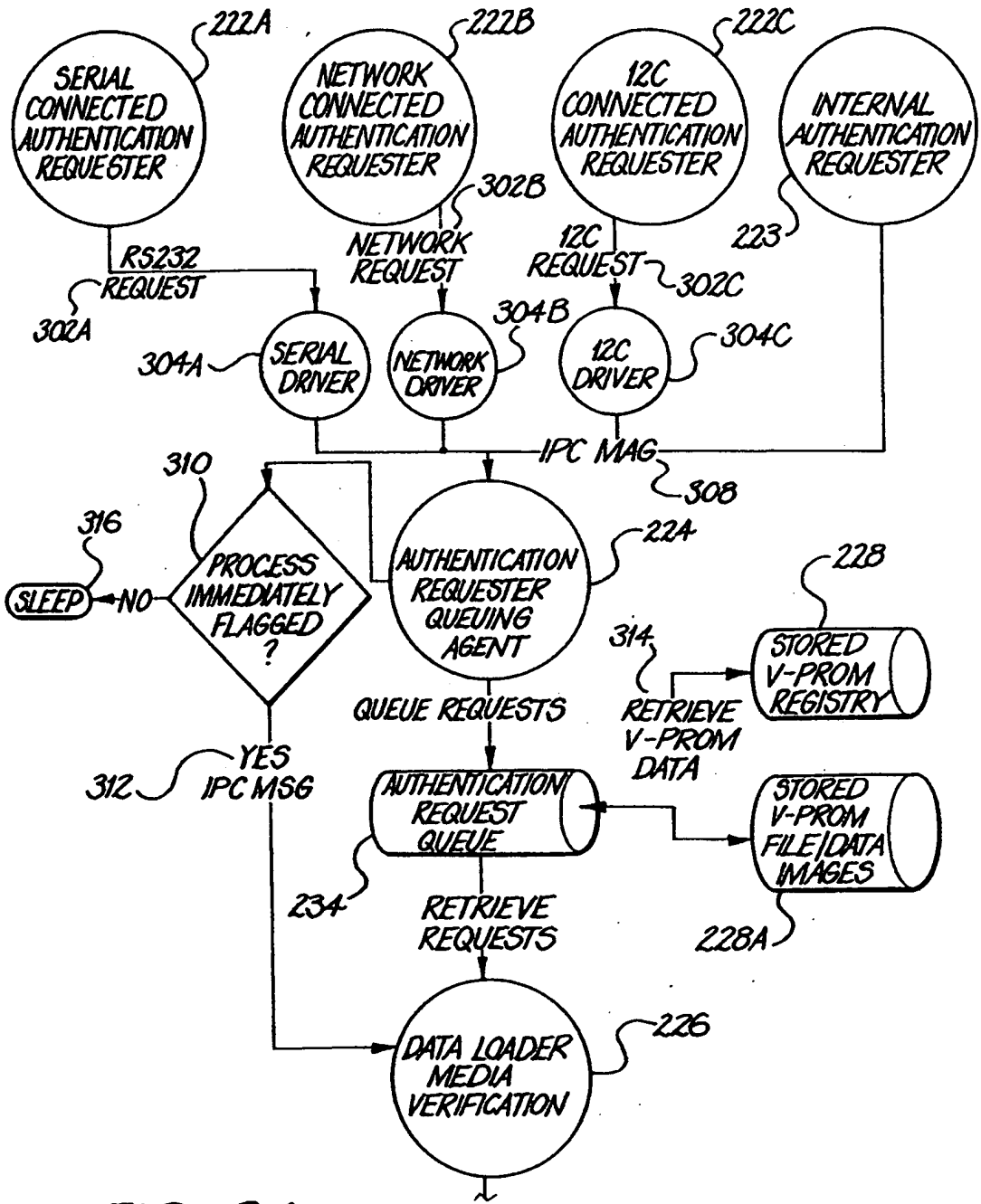

222A
SERIAL CONNECTED AUTHENTICATION REQUESTER
222B
NETWORK CONNECTED AUTHENTICATION REQUESTER
222C
12C CONNECTED AUTHENTICATION REQUESTER
INTERNAL AUTHENTICATION REQUESTER
223
302B
NETWORK REQUEST
12C REQUEST
302C
RS232 REQUEST
302A
304A
SERIAL DRIVER
NETWORK DRIVER
304B
12C DRIVER
304C
IPC MAG
308
310
316
SLEEP
NO
PROCESS IMMEDIATELY FLAGGED ?
AUTHENTICATION REQUESTER QUEUING AGENT
224
228
314
STORED V-PROM REGISTRY
RETRIEVE V-PROM DATA
QUEUE REQUESTS
312
YES IPC MSG
AUTHENTICATION REQUEST QUEUE
STORED V-PROM FILE/DATA IMAGES
234
RETRIEVE REQUESTS
228A
DATA LOADER MEDIA VERIFICATION
226

FIG. 3A

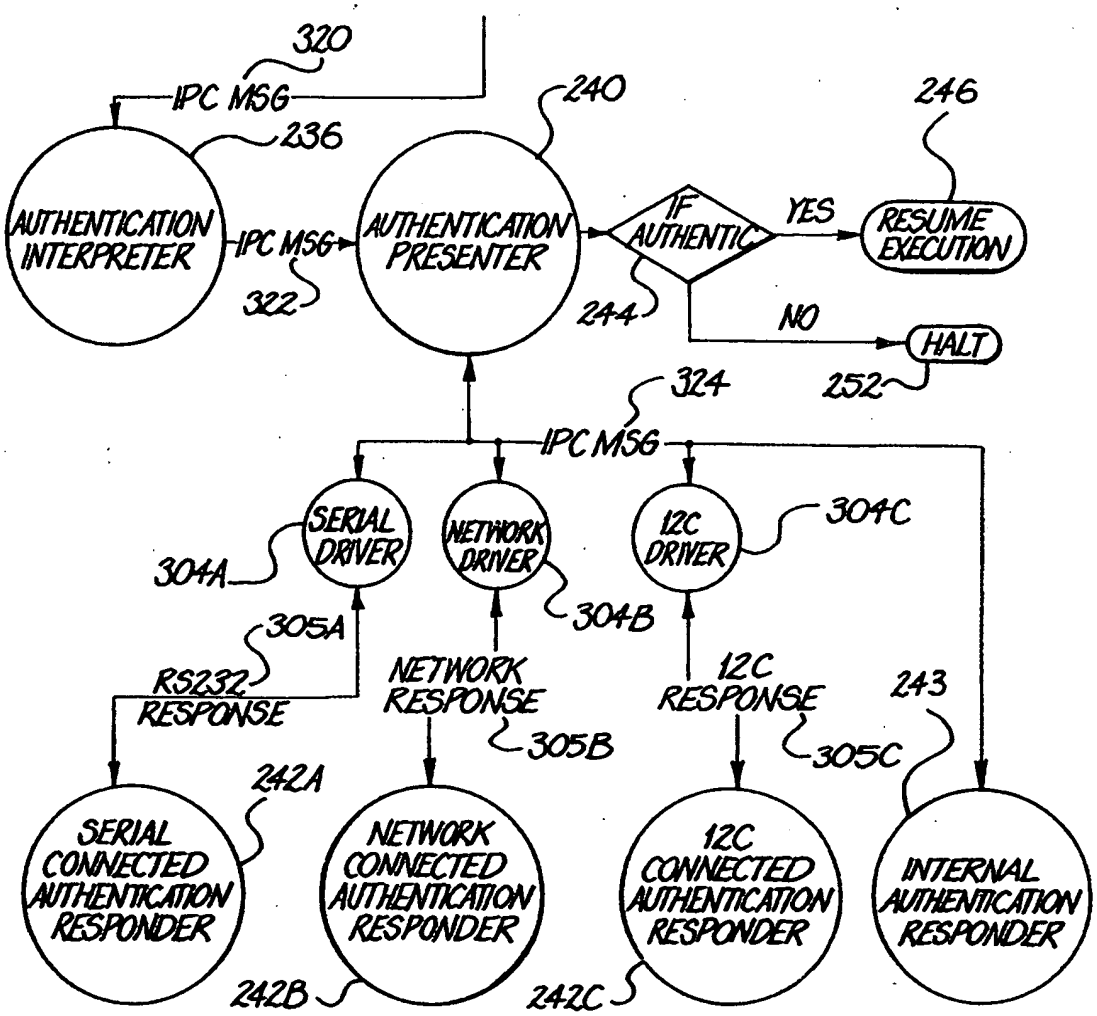

320
IPC MSG
240
246
236
AUTHENTICATION INTERPRETER
IPC MSG
AUTHENTICATION PRESENTER
IF AUTHENTIC
YES
RESUME EXECUTION
322
244
NO
HALT
252
324
IPC MSG
SERIAL DRIVER
NETWORK DRIVER
I2C DRIVER
304C
304A
304B
305A
RS232 RESPONSE
NETWORK RESPONSE
I2C RESPONSE
243
305B
305C
242A
SERIAL CONNECTED AUTHENTICATION RESPONDER
NETWORK CONNECTED AUTHENTICATION RESPONDER
I2C CONNECTED AUTHENTICATION RESPONDER
INTERNAL AUTHENTICATION RESPONDER
242B
242C

FIG. 3B